B ①

Bolt Beranek and Newman Inc.

REPORT NO. 3956

SPEECH COMPRESSION AND SYNTHESIS

QUARTERLY PROGRESS REPORT NO. 2
6 JULY - 5 OCTOBER 1978

OCTOBER 1978

PREPARED FOR:
ADVANCED RESEARCH PROJECTS AGENCY

Report No. 3956 Bolt Beranek and Newman Inc.

SPEECH COMPRESSION AND SYNTHESIS

Quarterly Technical Progress Report No. 2

6 July - 5 October 1978

ARPA Order No. 3515

Contract No. F19628-78-C-0136

Name of Contractor:
Bolt Beranek and Newman Inc.

Effective Date of Contract:
6 April 1978

Contract Expiration Date:
5 April 1979

Principal Investigators:
Dr. John Makhoul
(617)491-1850, x332

Dr. R. Viswanathan
(617)491-1850, x336

Sponsored by
Defense Advanced Research Projects Agency (DoD)
Monitored by RADC/ETC

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

TABLE OF CONTENTS

PROJECT PERSONNEL

John Makhoul	Principal Investigator
R. (Vishu) Viswanathan	Principal Investigator
Jared Wolf	Senior Scientist
John Klovstad	Senior Scientist
Lynn Cosell	Research Engineer
Richard Schwartz	Research Engineer
Dennis Klatt	Consultant
Victor Zue	Consultant
Peter Cudhea	Programmer
Kathleen Starr	Project Secretary

1. SUMMARY

During the past quarter, we have been working in the areas of natural phonetic speech synthesis, in real-time vocoder development, and in source modeling for multi-rate vocoders. Our progress in these areas is summarized below and described more fully in Sections 2 (speech synthesis), 3 (vocoder development), and 4 (source modelinq).

Speech Synthesis

During this quarter, we have brought the diphone speech synthesis program to the point where it produces synthetic speech from an input string consisting of triplets of phoneme identity, pitch, and duration, using diphone templates extracted from specially designed short utterances. We have done considerable experimentation with algorithms for diphone concatenation, time-warping, and interpolation, and we have also developed new algorithms for generating continuous parameter tracks for gain, voicing and pitch, and mixed-source model cutoff frequency. We have recorded a large data base of utterances for diphone templates and synthesis experiments, and we have manually labeled a portion of this data base.

Vocoder Development

During this quarter, we have successfully adapted to our PDP-11/AP-120B configuration the real-time LPC network vocoder developed by ISI. We have run the vocoder back-to-back, and we have held cross-country network conversations with ISI. We have also implemented RTFUD, an RT-11/FORTRAN/debugging environment for the AP-120B vocoder program. This program operates in non-real-time with files for input/output. It offers greatly improved debugging and statistics-gathering facilities, which have proved invaluable in localizing some problems with the vocoder and in making and testing vocoder algorithm modifications. During this quarter, we implemented two of the planned algorithm modifications to the real-time vocoder: the lattice-form synthesis filter and a more sophisticated variable frame-rate algorithm for the spectral (LPC) parameters.

Source Modeling

The goal of our work in source modeling is the development of improved vocoder excitation source models that result in more natural speech quality. In this context we are not only concerned with pitch-excited vocoders, but also residual- and voice-excited vocoders, at intermediate data rates (9.6 kb/s and below). We have developed and tested a new method for regeneration of the high-frequency portion of a residual excitation, given a transmitted baseband. In one of its forms, this method requires little or no extra computation.

2. PHONETIC SPEECH SYNTHESIS

During this quarter the synthesis programs were developed to the point where speech could be synthesized using only the input information necessitated by the very low rate (VLR) requirement; a sequence of phonemes with one duration and pitch value per phoneme. This is a significant step toward the realization of our proposed synthesis goals. Since then, much exploratory research has been done in an attempt to improve the quality of the synthesized speech.

We have organized this QPR section to emphasize known issues and our current implemented solutions. The first section provides a framework for the presentation of our work. It describes issues germane to all modeling work based on the use of template information. The second section presents our research as a systematic way of dealing with these issues. We show how our work has been influenced by an appreciation of the causes for the most noticeable speech quality problems. The third section describes our current synthesis results. In the final section we present our current view of where we go from here. This includes plans for continued improvement in speech quality as well as completing the diphone inventory.

2.1 Very Low Rate (VLR) LPC Speech Synthesis Issues

A fundamental design decision in this project was to characterize time-spectral energy patterns by diphone templates that have been extracted from real speech. A primary issue then is the acquisition of these diphone templates. This is discussed in Section 2.1.1 below.

Given a set of diphone templates, a second issue consists of mapping template information onto the input phoneme/duration string so as: (1) to satisfy the given duration requirements, and (2) to preserve (as much as possible) the naturalness of human speech. This is discussed in Section 2.1.2.

A third issue concerns the generation of a suitable excitation function from the given input (and diphone templates). This is discussed in Section 2.1.3.

2.1.1 Diphone Acquisition Issues

In our last QPR [1] we reported on our initial synthesis work in which sentences were synthesized using diphones taken from continuous speech. We now feel that using diphones that have been extracted from continuous speech is undesirable because: a) some are reduced and are therefore non-typical, b) some are too short, and c) there is a substantial problem in choosinq which instance (i.e., the phonetic context) of a particular diphone template to use in synthesis. Our research on diphone acquisition is described in Section 2.2.1.

2.1.2 Issues Concerning the Use of Diphone Templates

Having made the decision to use diphone templates as our source of time-varying spectral information, we must deal with issues concerning their proper use. These issues will be discussed in the following two subsections as time-warping and continuity issues. Our research on the use of diphone templates is described later, in Section 2.2.2.

2.1.2.1 Time Warping Issues

One issue concerns the time warping of template information (whose duration is fixed) in a way consistent with the durations specified by the input. What makes this difficult is that the time warping must preserve the naturalness of speech. This problem is constrained somewhat by the following three design choices:

1) The only timing information that is directly available as input is the duration of each phoneme to be synthesized. The required duration of any phoneme may, of course, change from one occurrence to the next.

2) In order to provide input to the LPC synthesis program we must specify LPC coefficients at fixed intervals (10 ms).

3) The duration of any particular diphone template is fixed once that diphone has been selected during diphone acquisition.

Since we are using template information as a source for time-varying parameters, we need a mapping function that defines what part of the template is to be used at each instant of time. We have constrained ourselves to piecewise linear mapping functions. The process of mapping template information onto continuous time can be viewed as a two-part procedure. First, we establish a correspondence between specified points in the template and the times at which those points in the template should be synthesized. This sequence of points is then used to define a piecewise linear *mapping function*. This correspondence and the resulting mapping function are illustrated in Figure 1. Notice that the template information is generally compressed during the mapping. The reason for this is that (whenever possible) we have intentionally chosen fully articulated diphone templates. These templates tend to be longer because they were obtained from specially chosen short utterances (and not continuous speech). Furthermore, we believe that compressing a long template will result in better speech quality than expanding a short one, since information is more easily ignored than generated.

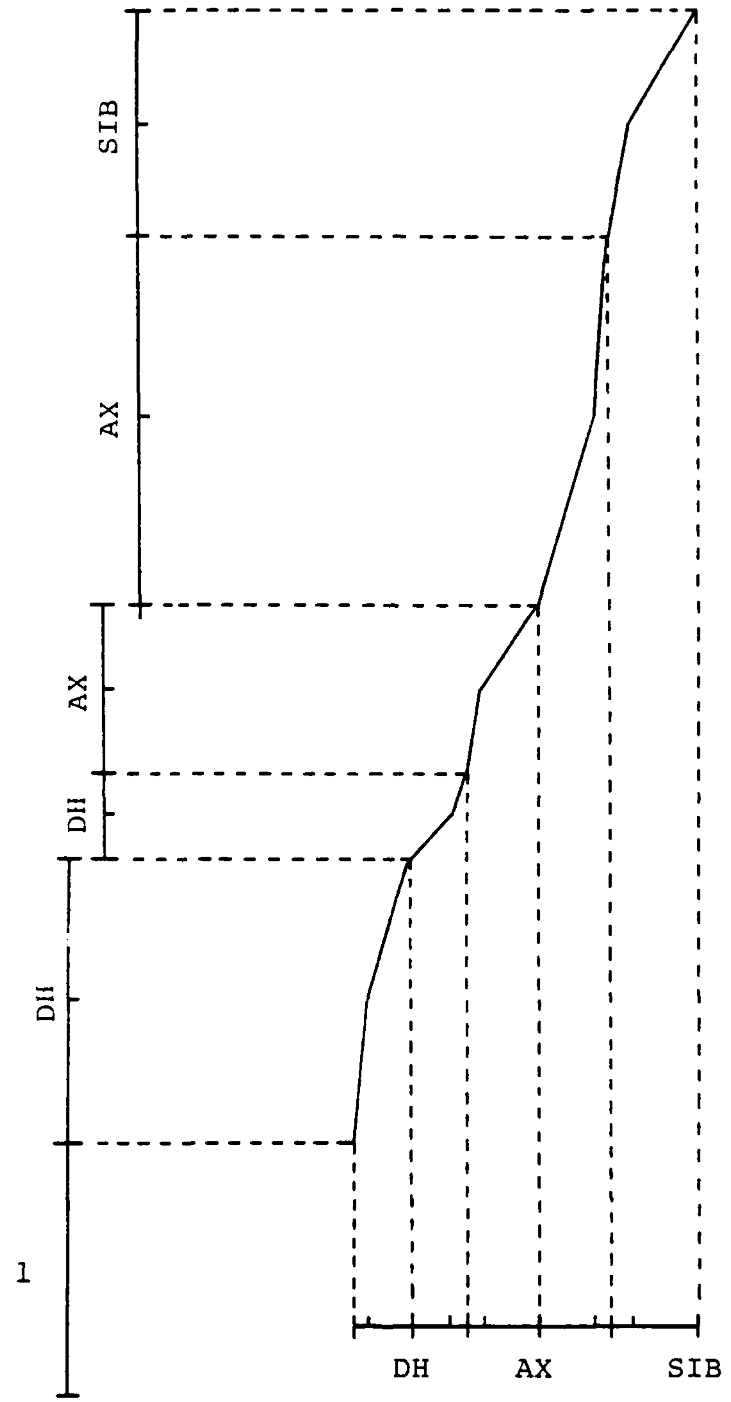


Fig. 1 Piecewise linear diphone mapping function.

2.1.2.2 Continuity Issues

Given a satisfactory mapping function (for template information), simple concatenation of the sequence of templates usually results in discontinuities at template boundaries. Therefore, a second issue concerning the use of template information is the prevention (or minimization) of unnatural discontinuities at template boundaries.

The concatenation of diphone templates involves the specification of both spectral and gain information. We have found that although both pitch and gain must be closely coordinated with the spectral information, gain is tied more closely to spectral characteristics (and template information). Therefore, gain is discussed along with spectral information as a continuity issue, whereas pitch is discussed separately.

2.1.3 Excitation Issues

The issues discussed above have dealt only with the specification of a time-spectral envelope that is consistent with the input sequence of phonemes and durations. There remains the problem of providing a natural sounding excitation for this time-varying spectral envelope. At a minimum, this requires the generation of pitch tracks and a determination of voicing mode within phonemes. Improvement beyond this requires a more complex excitation function. Previous work [2,3] has shown that our mixed-source model of excitation results in more natural sounding (less buzzy) speech at the cost of specifying a cutoff frequency with every pitch value. Section 2.2.3 describes our research on these excitation issues.

2.2 Attempts To Deal With These Issues

In this section we trace our research as it has evolved in a nearly chronological sequence. In order to explore various algorithms, we have installed each new variation as an option in the synthesis program.

2.2.1 Diphone Acquisition

After consideration of the problems inherent in using diphone templates that have been extracted from continuous speech (Section 2.1.1), we concluded that a more systematic approach to diphone template acquisition was needed. Two particularly attractive alternative approaches were developed.

One approach is to record a short word or phrase for every diphone needed. The word or phrase would be carefully selected to provide the correct phonetic context and stress pattern. The other approach is to record a large number of short nonsense utterances thatt have been systematically designed to include each of the basic diphones in a neutral phonetic environment and an advantageous stress pattern.

We knew that the first approach could be used to obtain any diphone whatsoever. We were also aware that a considerable amount of effort would have to be spent to specify the 2500 words or phrases that would be needed if we were to pursue that approach. However, since nonsense utterances could be systematically generated for any desired diphone, the second approach was especially desirable. Another advantage of the second method is that systematic generation of diphones results in a more uniform phonetic context. We did an experiment to see whether the diphones extracted from nonsense utterances could give good results. We concluded that they could indeed give consistent results and that they were much more practical than hand-picked words. In those few cases where diphones derived from nonsense utterances were not adequate, we have used one from a more appropriate word or phrase.

2.2.2 Use of diphone information

In this section we describe the algorithms that produce LPC parameter tracks for phonetic sequences with arbitrary duration. All of these algorithms are implemented and available in the synthesis program. The research is presented as it pertains to the two major template usage mentioned in Section 2.1.2: (a) time warping and (b) continuity across diphone boundaries.

2.2.2.1 Time Warpinq

We wish to describe some of the options currently available for defininq the piecewise linear mapping function (discussed in Section 2.1.1 and illustrated in Figure 1). The various options are (generally) independent and will therefore be discussed separately.

Selecting Diphone Boundaries in the Templates

Conceptually, there are three ways template diphone boundaries can be selected:

1) An experienced person can label the diphone boundaries in each template, or

2) The diphone boundaries can be selected to coincide with the midpoint between adjacent phoneme boundaries, or

3) Diphone boundaries can be selected dynamically as the frames (one in each of the diphone templates that are being abutted) whose spectra are most similar (according to some distance metric). Specifically, since the parametric information used to synthesize any particular phoneme is derived from the information in two diphone templates, frames near the phoneme middle in the left diphone template are compared to frames near the phoneme middle in the right diphone template.

The first option was implemented because it was thought that the person specifyinq diphones would be able to identify optimal diphone boundaries consistently. Because a great deal of time is required to do this, a program usinq the strategy described in the second option was implemented and found to give good results. Sometimes (with both of these options) the diphone boundaries appear to be correctly labeled, yet noticeable discontinuities exist across adjacent diphone boundaries (the middle of the phoneme). Because of this observation, the third option was implemented as a possible way of minimizing these spectral discontinuities in the middle of phonemes. Although this option is intuitively appealing, its use has not produced the expected improvement in synthesis quality. We conclude that either (1) there is no real improvement or, (2) the improvement is not very siqnificant and will become apparent only after more synthesized speech is available for comparison.

Selectinq Diphone Boundaries in the Synthesized Speech

The diphone boundaries in the synthesized speech are selected to be used in conjunction with those in the diphone template (cf. previous section) to uniquely define a point in the piecewise linear mapping function. The two ways currently implemented for selectinq these points are as follows.

1) The lengths of the two diphone halves (which are going to be used for the current phoneme) are added together. The ratio of the phoneme duration to this length is calculated and is used to define a scaling factor. Each diphone half is scaled by this scaling factor during the mapping process. The effect of this procedure is to pick a diphone boundary in the synthesized speech at a point that partitions the phoneme into parts with lengths proportional to the lengths of the diphone halves from which they are derived.

2) The midpoint of the phoneme is picked as the position for the synthesized diphone boundary. The result of this is that the diphone halves are very often scaled differently.

We believe that the second of these two diphone boundary selection methods will result in better speech quality because the diphone template boundaries that are going to be mapped onto this position originally corresponded to the middle of a phoneme. Our testing of this option has produced favorable (but not yet conclusive) evidence for its use.

Mapping Elastic and Inelastic Regions of the Template

The principle of distinguishing between "elastic" and "inelastic" regions of the template arises from observations of speech parameters under widely varying speaking rates. Most of the durational variation is observed to occur during the "steady state" portion of the phoneme (an elastic region), but the transition portions (an inelastic region) are relatively insensitive to changes in speaking rate. We were prompted to make such distinctions in an attempt to mimic this kind of behavior with synthesized speech. The synthesis program allows us to specify (via a parameter) that a certain percentage of the diphone template on each side of the phoneme boundary is to be treated as relatively inelastic and that the rest of the diphone template (the section correspondinq to phoneme middles) is to be treated as more elastic. The mapped duration of these regions is found by solving two equations. One equation relates the required duration of the sum of these two regions to the stretch factors used in warping each of the regions. A second equation relates the two stretch factors (via a parameterized relationship). Solving these two equations for the stretch factors permits a unique specification of the mapping function (for this phoneme). We intend to find (and permanently fix) the parameter values that yield the most natural simulation of this variable speaking rate behavior.

2.2.2.2 Continuity

When templates are concatenated, discontinuities across the template boundaries may verv well result even when the parameters are quite continuous within each template. In order to deal with this, we define an "interpolation region" that straddles the diphone template boundary (phoneme middle) and contains potential parametric discontinuities. This interpolation region is defined indirectly by labeling each diphone template with two "interpolation points". The right interpolation point specifies the left edge of an interpolation region. The right edge of this interpolation region is specified by the left interpolation point in the next adjacent diphone template.

The next two sections describe options (currently available in the synthesis program) that have been specifically designed to minimize spectrum and gain discontinuities across diphone templates by some kind of interpolation within the interpolation region.

Spectrum

We have implemented and tested the following different algorithms for preserving spectral continuity across template boundaries.

1) Linear Interpolation of the LAR Coefficients

We know from our work with variable-frame-rate transmission of log-area-ratio (LAR) spectral parameters, that large regions of parameters can be replaced by interpolated vectors of LAR parameters. LAR coefficients inside the interpolation region are calculated by simply interpolating between the two sets of LAR coefficients at the boundaries of the interpolation region. This kind of interpolation is illustrated in Figure 2a.

2) Adding a Ramp to LAR Coefficients

Since the LAR coefficients within the interpolation region are to come from the two different diphone templates being concatenated, we offset the coefficients in each template by a linear ramp in such a way that the discontinuity at the diphone boundary is eliminated. Each ramp is chosen so that the LAR coefficients at the boundaries of the interpolation region are unchanged. Since the ramps are linear, only one other constraint must be specified in order to uniquely define them (two points uniquely define a straight line). We have implemented this algorithm with two different kinds of constraints. One constraint is to have each of the ramps compensate for half of the discontinuity. This is illustrated in Figure 2b. The other constraint is to have each ramp compensate for a portion of the discontinuity that is proportional to the length of interpolation region. This is equivalent to saying that the slope of the two ramps must be equal. This is illustrated in Figure 2c. As can be seen in Figure 2, methods b and c often preserve information that is discarded by method a, resulting in more natural parameter transitions.

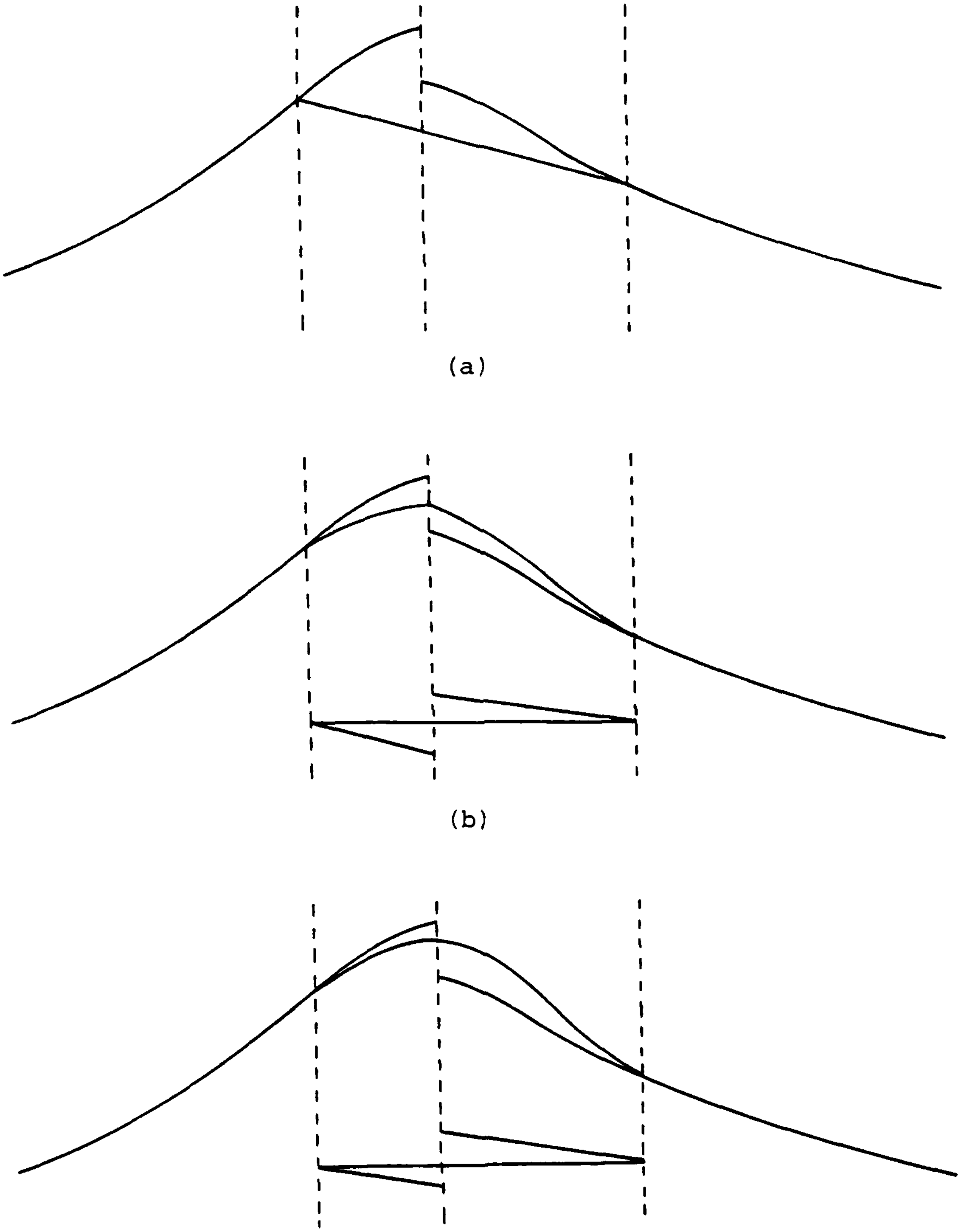


Fig. 2 Interpolation schemes

(a) Straight-line interpolation.
(b) Adding ramps with equal offset.
(c) Adding ramps with equal slope.

3) Low-Pass Filterinq of LAR Coefficients

We have implemented the capability to apply an arbitrary Finite Impulse Response (FIR) filter to the LAR coefficients. The intent for this option was to filter out the rapidly changing spectral discontinuities by low-pass filtering the LAR coefficients. All of the FIR filters that have been tried are zero-phase filters. Our experience so far indicates that a filter able to sufficiently reduce the parameter discontinuities also causes a noticeable slurring of the speech.

4) Interpolation Usinq Cumulative Energy Distributions

This kind of interpolation is the most complex of all the currently implemented interpolation schemes. The idea behind this scheme is to do the interpolation in a domain in which the results of the interpolation could be more clearly visualized. It was conceived of as beinq analogous to interpolation of formants. This interpolation consists of the following three major steps.

a) Calculate the cumulative spectral energy distribution for each of the two spectral frames between which an interpolated frame is desired. The cumulative energy distribution tells how much of the spectrum's energy occurs below any given frequency.

b) An interpolated energy distribution is calculated by interpolating between these two distributions. For each energy value (say 20% of the total energy in the spectrum) the two corresponding frequencies are found from their cumulative energy distributions. Then, the interpolation is done between these frequencies, resulting in a new interpolated frequency (below which 20% of the cumulative energy is to occur). The relationship between energy values and interpolated frequencies is sufficient to define an interpolated cumulative energy distribution.

c) The interpolated spectrum is derived by differentiating the cumulative energy distribution just calculated. This spectral interpolation can be used instead of LAR interpolation in each of the interpolation schemes just described.

Although the spectral interpolation using cumulative energy distributions is considerably more complex than any of the other interpolation methods and appears to behave as expected, it has not resulted in noticeably superior speech quality so far.

Gain

1) Template Gain

One option currently available for generating gain tracks is to use the gain tracks exactly as they appear in the diphone templates. The three interpolation methods that are illustrated in Figure 2 can also be applied to the gain tracks taken from the diphone templates.

2) Template Gain Plus Average Gain Targets

A second option is to correct each of the gain tracks from the diphone templates in a way that is consistent with "average" gain values in the phoneme middles. These "average" gain values are determined from the large inventory of diphone templates. The new gain track is generated by simply adding a linear component (with offset if necessary) to the gain track already in the diphone so that the observed average gain of each phoneme is achieved. This insures that there is no discontinuity at the diphone boundary.

The third interpolation option mentioned in 1), which adds a ramp to the gain track in the template (as illustrated for LAR's in Figure 2c) to eliminate discontinuities, results in the most natural speech quality.

2.2.3 Excitation

We have implemented several different excitation-related options. These options determine the way in which the single pitch value per phoneme is used to define a pitch track, which frames of the pitch track should actually be voiced, and how much frication should be used when voicing does occur.

1) Linear Pitch Tracks

A pitch track is created in the region between two consecutive pitch values by straight line interpolation.

2) Source Type Information

Source type information is used to determine whether each frame is to be voiced or not. If voicing is to occur, the pitch track is left unchanged; if voicing is absent, the pitch value for that frame is set to zero. Currently the source type information is determined by one or both of two ways: a) It can be taken directly from the diphone template, or b) it can be determined as a function of distinctive features of the phoneme being synthesized. For instance, all phonemes that have the binary feature of voicing (all vowels, semivowels, and voiced consonants) can be made to be voiced.

3) Mixed Source Model - Cutoff Frequency

With the mixed-source model [2,3], the excitation is composed of a mixture of voicing and frication such that the sum of their spectral envelopes is constant. The voicing excitation is low-pass filtered and the frication is high-pass filtered. The only degree of freedom in the mixed source model as described is the cutoff frequency (where the response is down 3 dB from the pass band) of the two filters. This frequencv marks the upper edge of the voicing spectrum as well as the lower edqe of the frication spectrum. Currently we have implemented an algorithm that selects a cutoff frequency on the basis of some of the phoneme's distinctive features: for vowels it is at 5000 Hz (fully voiced), for unvoiced consonants it is 500 Hz, and for voiced fricatives (which are produced with both periodic and random excitation) it is 1500 Hz. The cutoff frequency is low-pass filtered in order to minimize excitation discontinuities across phoneme boundaries.

Using straiqht-line interpolation between consecutive pitch values seems to produce satisfactorv pitch tracks. Making a voicing mode determination on the basis of certain phoneme features yields good results although we had thought that gettinq this information from the template on a diphone by diphone basis might be necessary. The implementation of the cutoff-frequency algorithm has resulted in a marked improvement in speech qualitv, in particular a decrease in buzziness of the synthesized speech.

2.3 Synthesis Results

We have recorded 100 of the Harvard Phonetically-Balanced sentences. Ten of these sentences have been hand-labeled and used extensively as target sentences in our current synthesis research. Each of the these 10 sentences has been synthesized in many different versions, corresponding to different sets of mappinq and/or smoothinq options in the synthesis program. We have compared these versions against one another in order to determine which of the synthesis options results in consistently better speech.

2.4 Future work

We view the future work to be done on this project as fulfillinq the two followinq goals: a) improvement of the quality of the synthesized speech, and b) extension of the diphone dictionary to permit synthesis of arbitrary English speech.

2.4.1 Qualitv Improvement

Since the most noticeable remaininq problem seems to be our handlinq of the gain we are planning to continue to improve our gain algorithm. It is quite possible that the remaining vowel loudness problems are all due to different vowel stress levels in the actual speech and can only be properly handled by transmitting vowel-specific stress information (about 1 bit/phoneme). However, we have not yet given up on the algorithms described above.

It is important to note here that the quality of the synthesized speech cannot be any better than vocoded speech. Our aim here is to approach that quality. Any further quality improvements must come from improving the quality of vocoded speech.

2.4.2 Completion of Diphone Inventory

The biggest remaining task is that of collecting the inventory of diphone templates necessary for the synthesis of unconstrained English speech. We still need to digitize, parameterize, and label each of the prospective diphones. We anticipate that this diphone acquisition process will continue throughout the remainder of the project even after the basic diphone inventory is complete, since one way to improve the quality is to use a different diphone template.

We feel that we have learned a great deal from the experiments performed thus far. We will be redesigning and recording a complete set of diphone utterances that will be better suited to the task of diphone synthesis. The initial digitization, parameterization, and labeling of these utterances should be a straightforward (though lengthy) effort.

3. REAL-TIME VOCODER DEVELOPMENT

3.1 Vocoder-Debugging Environment

As noted in the previous QPR [1], the PDP-11/AP-120B vocoder programs obtained from ISI contain no provision for file-to-file operation of the vocoder. Such a facility is desirable for testing the vocoder with a repeatable input (such as a digitized speech waveform from a file) and for observing the vocoder output in detail (as with a waveform file display program). Furthermore, even though ISI's FUD debugging program contains provisions for breakpointing the AP-120B program, the real-time structure of FUD prevents effective debugging of the running vocoder program in the AP-120B.

Because FUD presented such an unwieldy environment for the algorithm developments planned for the vocoder, we implemented RTFUD, a RT-11/FORTRAN/debugging environment for the AP-120B vocoder program. RTFUD operates the vocoder in non-real-time, using files for input and output of digitized speech waveforms and/or transmission parameters. Non-real-time operation implies that the vocoder can be stopped, examined, and then continued. Furthermore, the implementation in FORTRAN has permitted the rapid implementation of a number of statistics-gathering and diagnostic tools. RTFUD is used with AP FDT, a FORTRAN debugging program (FDT) that we have modified to permit it to debug the AP-120B as well as the PDP-11 program. The facilities of RTFUD have uncovered several bugs and problems in the AP-120B vocoder program, and they have been used for developing the variable frame rate (VFR) and synthesizer modifications described in Section 3.2 below. This section gives a brief description of the facilities offered by the current implementation of RTFUD.

Figure 3 illustrates the relation of the real-time AP-120B program to the RTFUD implementation. In the real-time implementation, the AP-120B vocoder program is organized as a closed loop. In RTFUD, the analysis (XANAL) and synthesis (XSYNTH) portions of the program are separately and explicitly controlled by the RTFUD program (bv means of APEX, the FPS-supplied "AP-120B Executive") in the PDP-11. The analysis and synthesis portions of the vocoder programs are effectivelv identical in both cases. Therefore, developments made to the AP-120B programs in the non-real-time configuration are directly transferable to the real-time system; RTFUD does not compromise the real-time goals of the vocoder development project.

The basic mode of operation of the RTFUD vocoder is "back-to-back", with digitized speech input from a file processed by the analysis and synthesis portions of the vocoder, producing synthesized speech output to a file. RTFUD also provides a real-time playback facility, which converts a disk file to speech

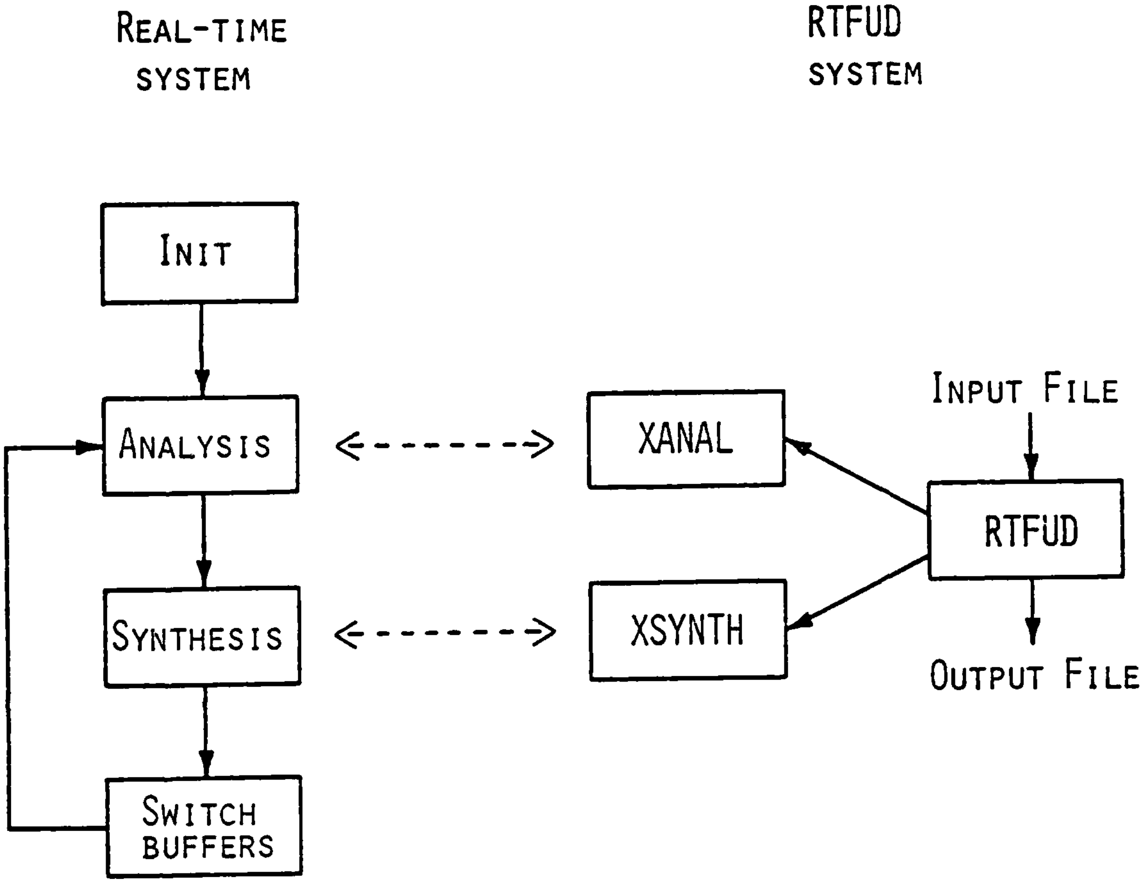


Fig. 3 Comparison of the AP-120B vocoder program in the real-time system and in the RTFUD system.

for listening purposes. RTFUD also provides for "analysis" and "synthesis" modes of operations, in which the respective output or input is a file containing transmission data (which in an actual real-time system would be transmitted through the communications network).

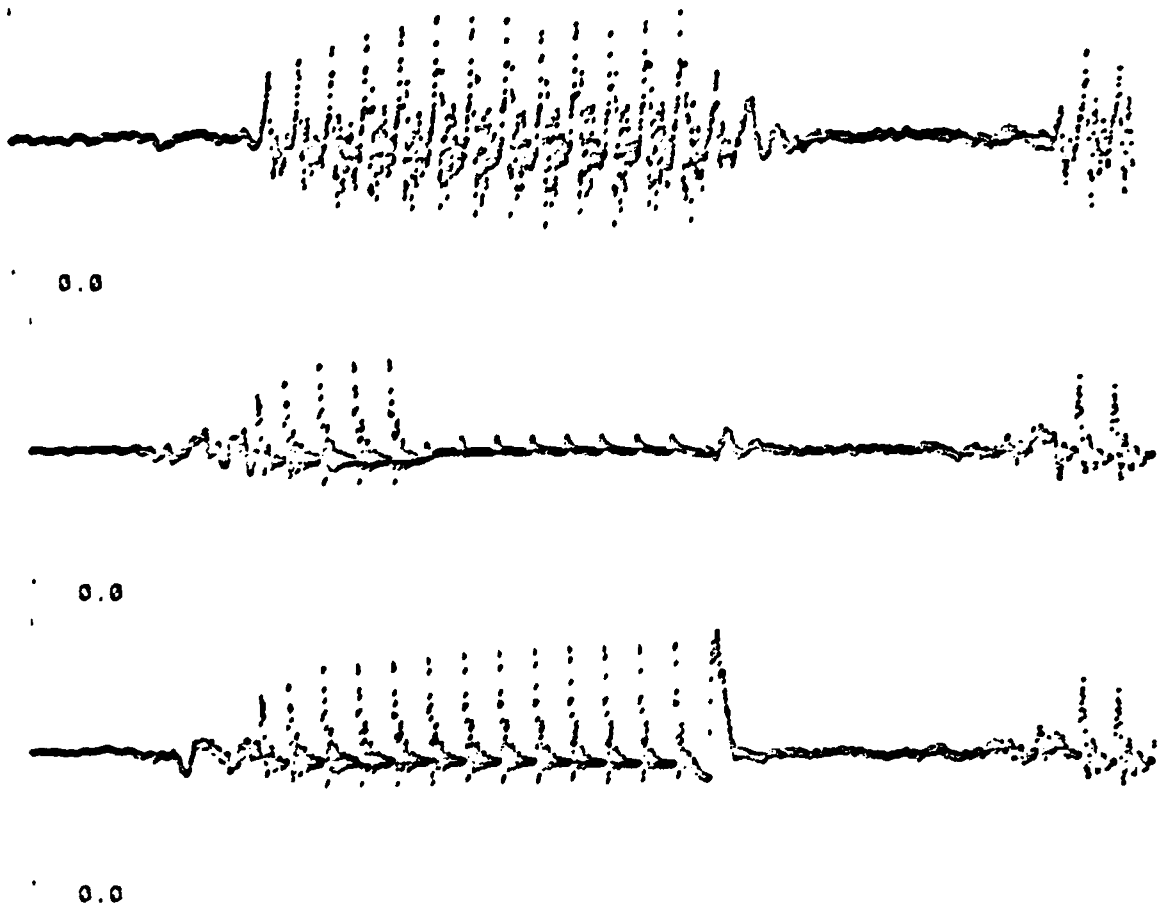
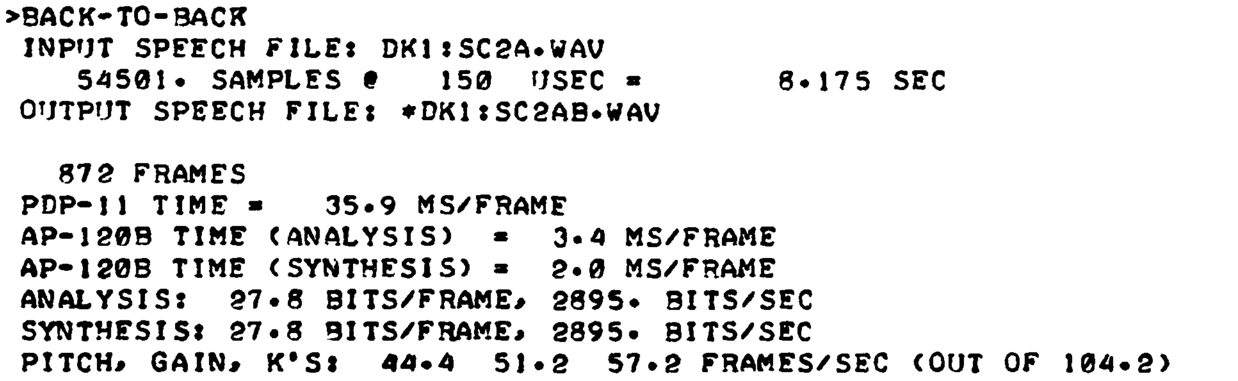


Fig. 4 Waveforms of the syllable "mass" in "Massachusetts": (top) original; (center) vocoded with gain bug; (bottom) vocoded with gain bug fixed.

Figure 4 illustrates the value of being able to use a known, repeatable input to the vocoder and being able to observe the input and output waveforms in detail. The figure shows three versions of the waveform of the syllable "mass" from the word "Massachusetts". The top trace is the original digitized speech; the center trace shows the same syllable at the output of the ISI vocoder. A program bug has inappropriately lowered the gain about half-way through the syllable. By editing the input file to contain just this syllable, and observing certain vocoder memory locations after each frame of speech, it was an easy task to localize and correct this bug in the synthesis interpolation routine. The output speech after this bug was fixed is shown in the bottom trace of Figure 4.

RTFUD accumulates several statistics of the vocoder, which can be valuable in understanding and adjusting the operation of the vocoder. Figure 5 illustrates the type of statistics currently printed.



```
>BACK-TO-BACK
 INPUT SPEECH FILE: DK1:SC2A.WAV
    54501. SAMPLES @   15Ø  USEC =        8.175 SEC
 OUTPUT SPEECH FILE: *DK1:SC2AB.WAV

   872 FRAMES
 PDP-11 TIME =   35.9 MS/FRAME
 AP-12ØB TIME (ANALYSIS)  =  3.4 MS/FRAME
 AP-12ØB TIME (SYNTHESIS) =  2.Ø MS/FRAME
 ANALYSIS:  27.8 BITS/FRAME, 2895. BITS/SEC
 SYNTHESIS: 27.8 BITS/FRAME, 2895. BITS/SEC
 PITCH, GAIN, K'S:  44.4  51.2  57.2 FRAMES/SEC (OUT OF 1Ø4.2)
```

Fig. 5 Statistics summary output by RTFUD.

RTFUD contains provisions for accumulating histogram data on the transmitted parameters. Figure 6 shows a histogram display derived from vocoding about 1Ø sec of speech from one (male)

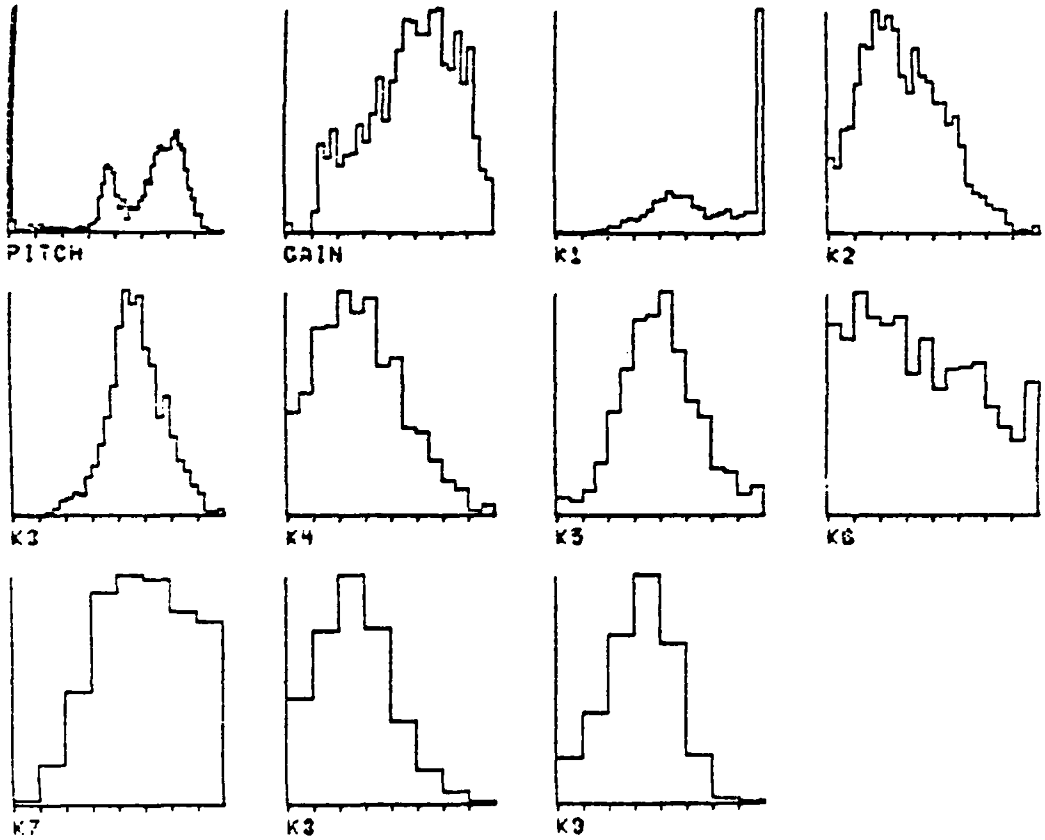
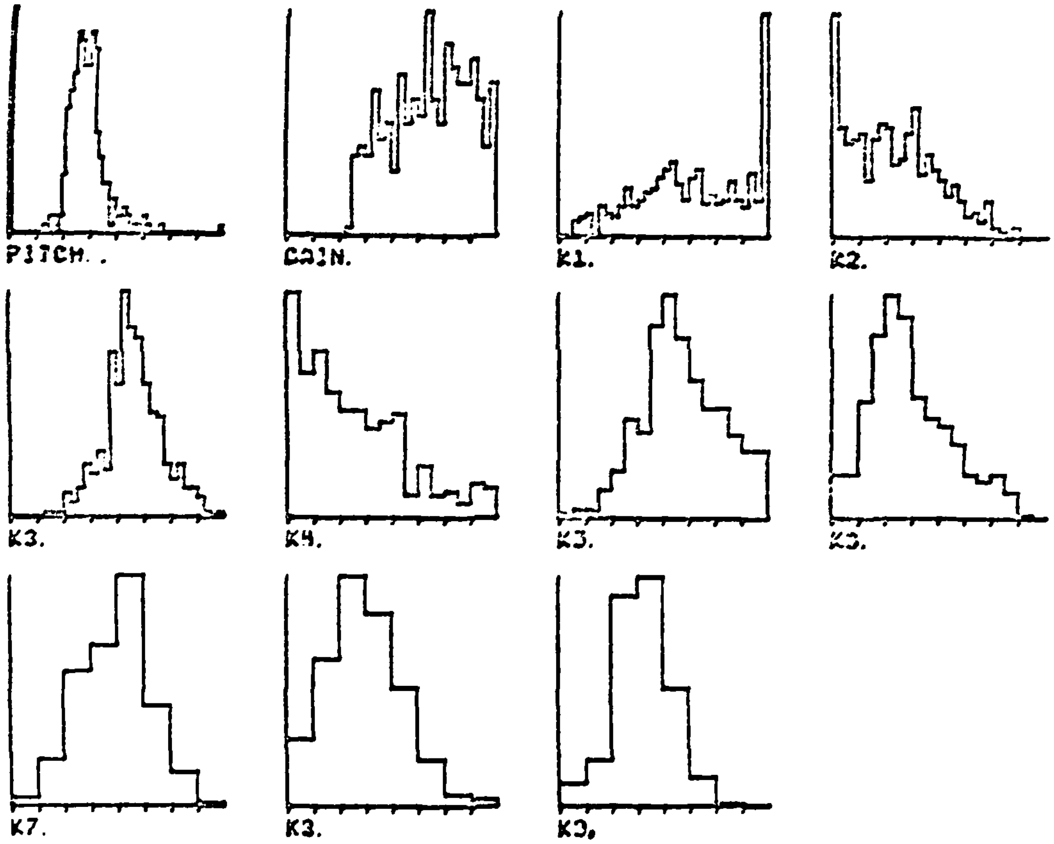


Fig. 6 Histogram display of transmitted parameters, single (male) speaker.

speaker. (The histogram display was made with IMSYS, the graphics system implemented on our PDP-11 during the previous quarter [1].) Several valuable observations about the vocoder operation may be made from this figure. Note that the pitch period histogram is distinctly bimodal. The right-hand lump represents correct pitch periods for this speaker, and the left-hand lump represents octave errors (pitch periods of half the correct period). Also note the K1 (first reflection coefficient) histogram. It shows that the highest coded value of K1 is used a disproportionate amount of the time. In other words, the quantization tables appear not to encompass the full range of K1 computed by the analysis portion of the vocoder. Figure 7 shows a comparable histogram for another



Fiq. 7 Histogram displav of transmitted parameters, single (female) speaker.

(female) speaker. The pitch period histogram shows shorter pitch periods, as expected, and few (if any) octave errors. The K1 histogram shows the same effect as before, and for this speaker, K2 and K4 are also severely skewed. These effects will be investigated in the coming quarter.

RTFUD contains provisions for producing frame-by-frame listings of various vocoder parameters such as analysis parcels (before VFR decisions), synthesis parcels (after VFR decisions), and unquantized reflection coefficients (to be used in investigating the quantization table problems discussed in the previous paragraph). Because the FORTRAN environment of RTFUD facilitates such formatted output, during the testing of the new VFR algorithm it was a simple matter to add a special printout of the VFR tables so that the operation of the VFR algorithm could be monitored and verified.

RTFUD also contains commands for controlling such things as enabling or disabling VFR operation of the vocoder, controlling double-threshold pitch and gain transmission (not previously implemented), and adjusting the VFR thresholds for the desired transmission rate.

3.2 Real-Time Vocoder Modifications

3.2.1 Establishing Current Version

In early July, after having collected the necessary source code modules from the NSC Group at ISI and adapting them for our configuration, we successfully tested a real-time, non-network, back-to-back implementation of the LPC vocoder. During late July, we brought up the network version of the vocoder and held our first network conversation. Since then, several configuration-dependent bugs have been found and fixed. At least one bug remains to be located and fixed. The symptoms of the bug, an almost-regular popping noise in the synthesized speech output, manifest themselves only under the network version of the vocoder, and only in the synthesis portion of the vocoder running at BBN (not at ISI). The symptoms appear whether we are receiving speech from another site or we are running in loop-back mode, independent of the network. We have further determined that the bug manifests itself only when we are performing VFR analysis. Since analysis and synthesis are independent processes, we believe that the problem is possibly one of memory mapping incorrectly performed by EPOS in our configuration. We are continuing to consult with ISI on finding and correcting this bug.

3.2.2 New Variable-Frame-Rate Modification

We have implemented and are in the process of testing the new VFR algorithm described in [2]. We have successfully modified the vocoder to use this new algorithm.

The new VFR algorithm operates by modeling what would occur in the synthesis process if the current packet were transmitted. If the average difference between the actual LAR values since the last transmitted frame and the interpolated values is less than a threshold (implying that linear interpolation between the end points is satisfactory), no transmission occurs. If the average distance is greater than the threshold, implying that the current frame is not part of the previous trend, the previous frame is transmitted. Thus, the new VFR algorithm finds suitable end points for line segments that reasonably approximate the computed LARs. These end points are then transmitted.

We are currently experimenting with threshold values and other parameters in order to produce the best quality synthesized speech at a lower transmission rate.

3.2.3 Lattice Synthesis

We have also modified the vocoder to use a lattice-form synthesis filter instead of the normalized form that was previously used. The normalized-form filter was advantageous in the SPS-41 vocoder implementation, but not in the AP-12ØB. The lattice-form requires only two multiplies per stage, compared to four for the normalized-form. We used the computation time saved by this reduction to make the synthesis routine more modular. Where before, the arguments to the routine were hand compiled and were an integral part of the design of the vocoder, they are now passed to the subroutine and can be general.

The normalized-form filter required the arc sine of the reflection coefficients, used as pointers to the filter coefficients. Decoding of transmitted parameters and linear interpolation were performed in this arc sine domain. The lattice-form filter uses the reflection coefficients themselves. However, since the new VFR algorithm assumes linear interpolation in the LAR domain (for reasons of spectral sensitivity), we modified the vocoder synthesizer to decode and interpolate in this domain. We then added a subroutine to convert from LARs to reflection coefficients for the synthesis filter computations.

4. SOURCF MODELING

4.1 A Residual-Based Source Model

One type of voice-excited coder transmits a low-frequency band of the residual (up to 1000 Hz), known as the baseband, and uses it as a basis to derive an excitation signal at the receiver. The general method used is to rectify the baseband, then spectrally flatten it in order to regenerate the high frequencies of the excitation signal. This high-frequency regeneration (HFR) method has been essentially the only one used thus far. While the method works fairly well, it requires a substantial amount of computation. We have developed a HFR method that, in one of its forms, requires little or no computation. The new method can be viewed as a model that determines exactly the high frequencies, given the baseband.

The idea behind the new method derives from the pitch-excited coder. In voiced excitation, the spectrum of the excitation is a flat line spectrum at multiples of the fundamental pitch frequency. Such a spectral structure is periodic and repetitive: the high-frequency structure is the same as at low frequencies. The spectrum of unvoiced excitation, on the other hand, is continuous and has a random spectrum with a flat envelope. However, the details of the unvoiced spectrum are not as important perceptually as the details of the voiced spectrum. Therefore, the unvoiced spectrum can be considered repetitive also, in that any similar spectrum can be substituted with equally good results.

The new regeneration method, then, is simply to duplicate the baseband spectrum at higher frequencies, in some fashion. We shall show how this may be done by the aid of Figures 8 and 9.

We shall assume that we are in the context of a predictive coding system. From Figure 8a, the speech signal s(t) of bandwidth W Hz is inverse filtered to obtain the residual e(t). The residual is then lowpass filtered at B Hz, where B<W, and decimated to a sampling frequency of 2B Hz. The resulting baseband signal v(t) is then quantized and transmitted. (For the moment, we shall neglect quantization in our discussion.) Figures 9a, 9b, and 9c show idealized spectra of the residual signals e(t), e'(t), and v(t), respectively. The hatched areas denote the region below the Nyquist frequency. The direction of hatching of negative frequencies is the mirror image of that for positive frequencies to indicate that the spectra are symmetric (folded) about the vertical zero axis.

4.2 Spectral Folding

Figure 8b shows the receiver of the proposed system. The received baseband residual is interpolated to 2LB Hz by inserting L-1 zeros after each residual sample, where L is the smallest integer larger than or equal to W/B. The interpolated signal x(t) is then lowpass filtered at W Hz, decimated to 2W Hz, amplified by W/B for energy adjustment, and used as the excitation u(t) to the

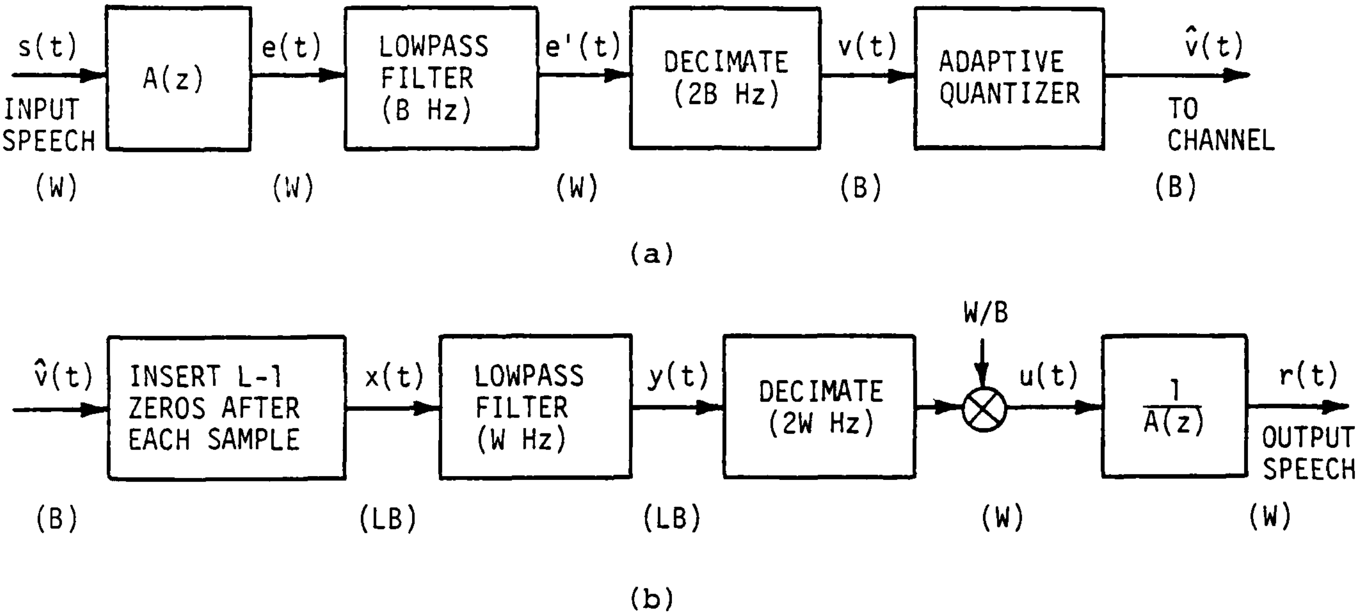


Fig. 8 Baseband residual high-frequency regeneration general scheme by spectral folding in a voice-excited coder. W/B is arbitrary.

(a) Transmitter.

(b) Receiver.

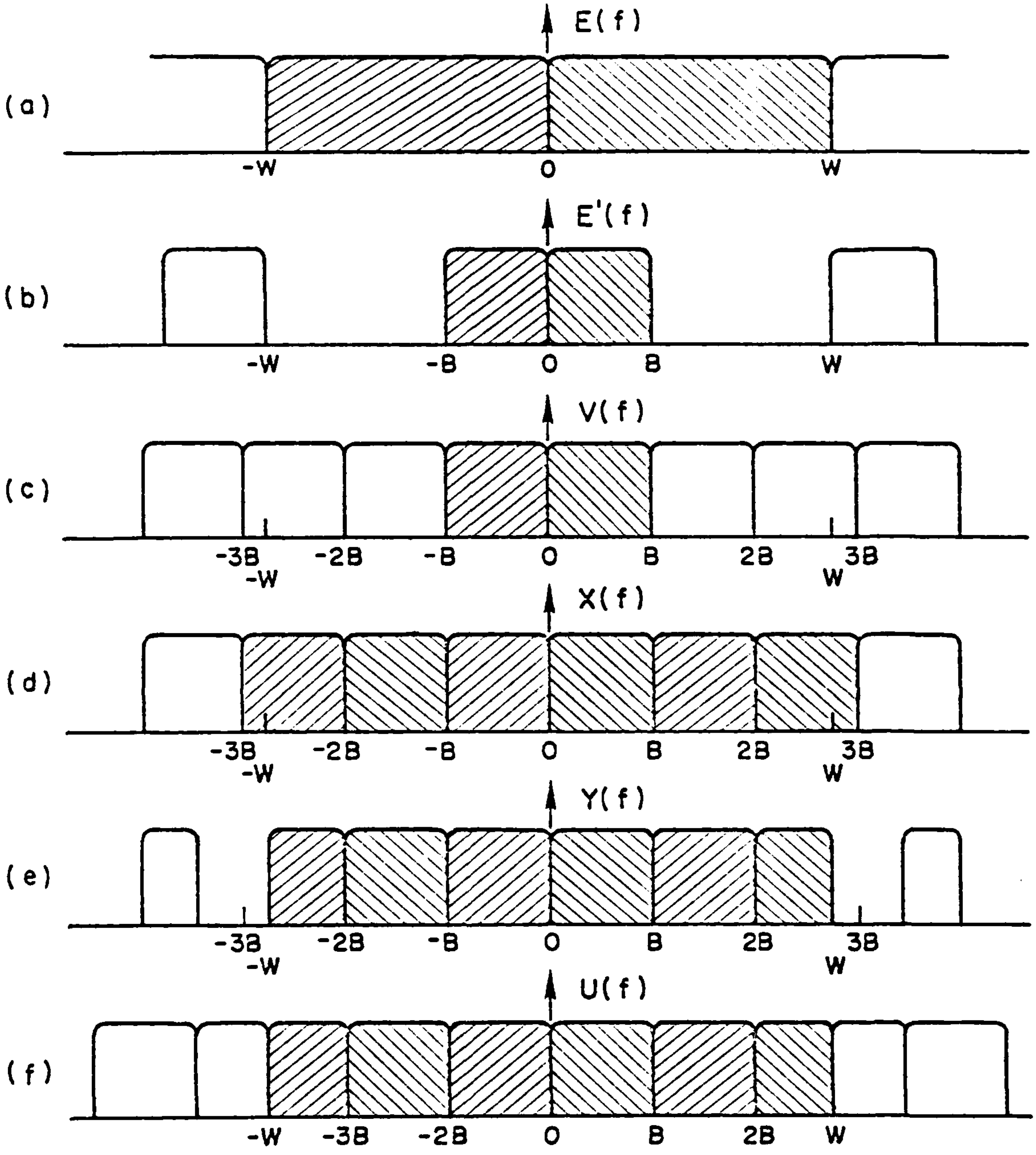


Fig. 9 Amplitude spectra of the signals at the different points in Fig. 8 for a noninteger value of W/B.

synthesizer filter. Figures 9d, 9e, and 9f show the spectra corresponding to the siqnals x(t), y(t), and u(t), respectively. In this example, L=3.

Fiqures 8 and 9 are general in that B may be any arbitrary frequencv less than W. For the special, but important case, when W is an integer multiple of B, i.e., W=LB, Figure 8 can be realized in a much simpler wav as shown in Figure 1Ø. Note how the decimator in Figure 8a simplifies in Figure 1Øa to simply retaininq every Lth sample and discardinq the rest. Also, the lowpass filter and the decimator in Fiqure 8b are completely eliminated in the receiver of Fiqure 1Øb. Fiqure 11 shows the residual spectra of interest for L=2, i.e., the width B of the baseband is exactly one-half the residual bandwidth W. Note that the high-frequency portion of the excitation spectrum U(f) in Figure 11d is simply an aliased (folded) duplication of the baseband spectrum. For arbitrary L, one can easily see that the excitation spectrum in the frequency regions between 2mB and (2m+1)B, where m=Ø,1,2,..., will be identical to the baseband spectrum, while the intervening reqions will have a spectrum that is a folded version of the baseband spectrum. One can think of the process as that of spectral folding, where the region between mB and (m+1)B, m>Ø, is obtained by foldinq over the precedinq region between (m-1)B and mB. Therefore, we shall call this HFR method the spectral folding regeneration method.

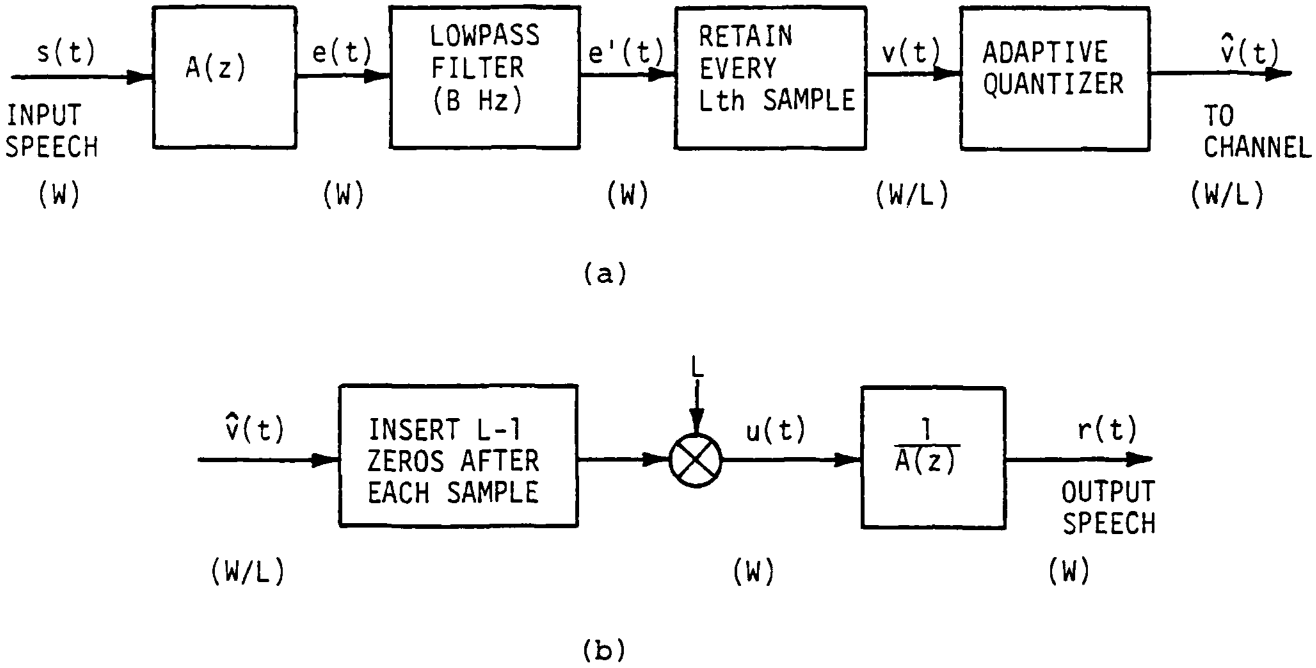


Fig. 10 Baseband residual high-frequency regeneration by spectral folding when $W/B = L$ is an integer.

(a) Transmitter.

(b) Receiver.

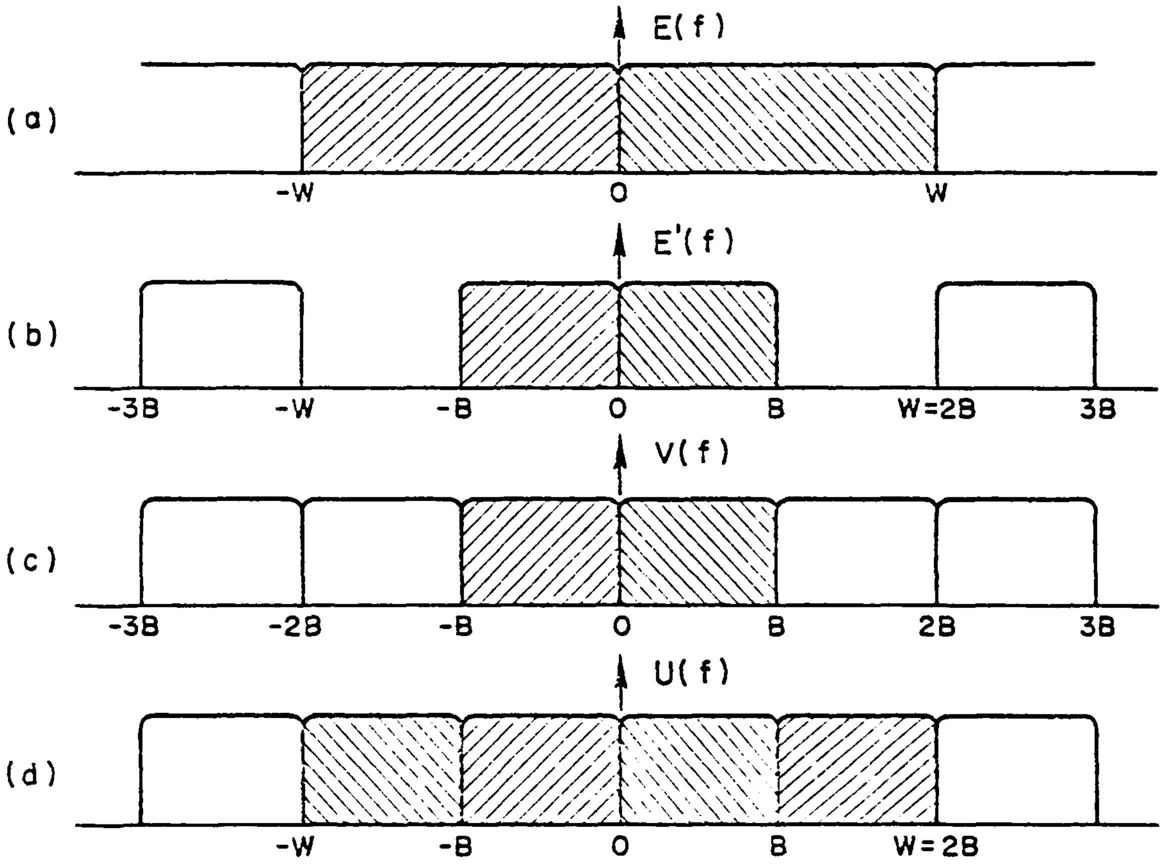


Fig. 11 Amplitude spectra of the signals at the different points in Fig. 10 for L=W/B=2.

4.3 Spectral Repetition

It is also possible to duplicate the baseband spectrum at high frequencies without any spectral folding. The process requires the use of heterodyning and further bandpass filtering. However, for the special case of L=2, the process may be greatly simplified. Figure 12 shows a schematic of the regeneration process in this special case. The top branch of the figure gives the baseband at the signal sampling frequency. The bottom branch generates the high frequency band. The multiplication of the input baseband residual by $(-1)^t$, i.e., changing the sign of every other sample, reverses the spectrum of the baseband, so that the high frequency region is now identical to the original baseband spectrum. In Figure 12, one need design only the lowpass filter; the highpass filter is simply obtained by reversing the sign of every other coefficient in the lowpass filter. The system in Figure 12 is only applicable for L=2. However, the design of systems for other values of L is straightforward. In contrast to the spectral folding method, we shall call this method the spectral repetition regeneration method.

4.4 Preliminary Results

It is clear from the preceding discussion that spectral folding with an integral number of bands, i.e., integer W/B, is the simplest of all HFR methods. We have tested its performance and

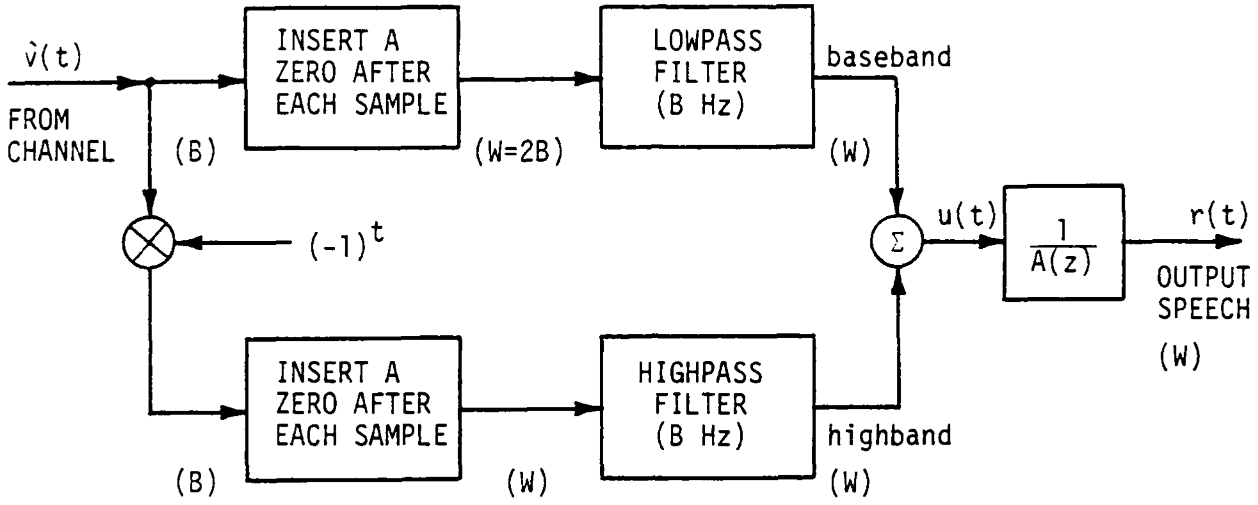


Fig. 12 The receiver portion of a voice-excited coder, showing high-frequency regeneration of the baseband residual by spectral repetition. This figure is for the special case when W/B = 2.

found the speech to be of good quality, with a minimal amount of distortion. The distortion is more noticeable with female speech, and may be due to the fact that the baseband does not generally contain an integral number of pitch harmonics.

5. REFERENCES

1. J. Wolf, L. Cosell, J. Klovstad, J. Makhoul, and R. Schwartz, "Speech Compression and Synthesis," Quarterly Technical Progress Report No. 1, 6 April - 5 July 1978, Report No. 3896, Bolt Beranek and Newman Inc., Cambridge, Mass., July 1978.

2. R. Viswanathan, J. Makhoul, and A.W.F. Huggins, "Speech Compression and Evaluation," Report No. 3794, Bolt Beranek and Newman Inc., Cambridqe, Mass., April 1978.

3. J. Makhoul, R. Viswanathan, R. Schwartz, and A.W.F. Huggins, "A Mixed-Source Model for Speech Compression and Synthesis," 1978 IEEE Int. Conf. on Acoustics, Speech, and Signal Processinq, Tulsa, Okla., April 10-12, 1978, pp. 163-166.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
BBN Report No. 3956

2. GOVT ACCESSION NO.
A155 416

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
SPEECH COMPRESSION AND SYNTHESIS

5. TYPE OF REPORT & PERIOD COVERED
Quarterly Technical Report
6 July to 5 October 1978

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(*s*)
Jared Wolf, John Klovstad, Richard Schwartz, Lynn Cosell, John Makhoul, Michael Berouti

8. CONTRACT OR GRANT NUMBER(*s*)
F19628-78-C-0136

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Bolt Beranek and Newman Inc.
50 Moulton St.
Cambridge, MA 02138

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Deputy for Electronic Technology (RADC/ETC)
Hanscom Air Force Base, MA 01731
Contract Monitor: Mr. Caldwell P. Smith

12. REPORT DATE
October 1978

13. NUMBER OF PAGES
52

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3515.

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Speech synthesis, phonetic synthesis, diphone, LPC synthesis, vocoder, speech compression, linear prediction, voice-excited coder, high-frequency regeneration.

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This document reports progress in the development of a phonetic speech synthesis algorithm, implementation and development of a real-time LPC vocoder, and development of a new high-frequency regeneration method for the excitation signal of voice-excited coders. The speech synthesis system now produces speech using an input string consisting only of triplets of phoneme identity, pitch, and duration, as required for the

cont'd.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

20. Abstract (cont'd.)

goal of very low transmission data rate. In real-time vocoder development, the implementation of an effective vocoder testing and debugging program and the implementation of the first two planned vocoder algorithm modifications are reported. The high-frequency regeneration method for voice-excited coders produces quite good voice quality at the cost of little or no extra computation.